This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1 (currently amended). A method of facilitating the use of a software process with one of a plurality of secure repositories, said method comprising the acts of:

providing an interface, said interface being callable by said software process;

if said one of said plurality of secure repositories is said first of said plurality of secure repositories, providing a first set of computer-executable instructions which are invocable by said callable interface; and

if said one of said plurality of secure repositories is said second of said plurality of secure repositories, providing a second set of computer-executable instructions which are invocable by said callable interface, said second set of computer-executable instructions being different from said first set of computer-executable instructions,

wherein said first of said plurality of secure repositories comprises a software module that uses a cryptographic algorithm to apply a cryptographic key to data without said cryptographic key being stored in a memory accessible to said first of said plurality of secure repositories, and

a' wherein said second of said plurality of secure repositories comprises a hardware module that uses said cryptographic algorithm to apply said cryptographic key to data, said hardware module further comprising hardware that resists or prevent divulgence of said cryptographic key outside of said hardware module.

2 (currently amended). The method of claim 1, wherein said one of said secure ~~repository~~ repositories converts encrypted data to decrypted data by using [[a]] said cryptographic algorithm to apply [[a]] said cryptographic key to said encrypted data, and wherein said software process performs an operation on said decrypted data.

3 (original). The method of claim 2, wherein said operation comprises rendering said decrypted data.

4 (original). The method of claim 1, wherein said first or said second sets of computer-executable instructions is provided in the form of an executable file dynamically linkable with said software process.

5 (original). The method of claim 1, wherein said interface comprises a first function callable by said software process, said first function being parameterized by first data representative of a type of secure repository.

6 (original). The method of claim 5, wherein said interface is callable by said software process without regard to whether said one of said plurality of secure repositories is said first of said plurality of secure repositories or said second of said plurality of secure repositories.

7 (original). The method of claim 1, wherein said interface comprises a second function callable by said software process, said second function requesting that said secure repository perform at least one action.

8 (original). The method of claim 1, wherein said first of said plurality of secure repositories executes on a closed-platform device, and wherein said second of said plurality of secure repositories executes on an open-platform device.

9 (currently amended). A method of communicating between a software process and a one of a plurality of secure repositories, said method comprising the acts of:

said software process issuing a first interface call which authenticates said software process to said one of said plurality of secure repositories; and

said software process issuing a second interface call which requests performance of an action by said secure repository for said software process;

wherein said software process issues said first and second interface calls without regard to whether said one of said plurality of secure repositories is a first of said plurality of secure

repositories or a second of said plurality of secure repositories, wherein said first of said plurality of secure repositories comprises a software module that uses a cryptographic algorithm to apply a cryptographic key to data without said cryptographic key being stored in a memory.

10 (currently amended). The method of claim 9, wherein said secure repository converts encrypted data to decrypted data using [[a]] said cryptographic algorithm to apply a cryptographic key to said encrypted data, and wherein said software process performs an operation on said decrypted data.

11 (original). The method of claim 10, wherein said operation comprises rendering said decrypted data.

12 (currently amended). The method of claim 9, wherein said first secure repository comprises a software-based secure repository, and wherein said second secure repository comprises at least some isolated hardware that uses said cryptographic algorithm to apply said cryptographic key to data and that resists or prevent divulgence of said cryptographic key outside of said second secure repository.

13 (original). The method of claim 9, wherein each of said first and second secure repositories are software-based repositories, said first secure repository having at least one feature not present in said second secure repository.

14 (original). The method of claim 9, wherein said one of said plurality of secure repositories is said first of said plurality of secure repositories, and wherein said software process issues said first and second interface calls without regard to whether said second repository exists.

15 (original). The method of claim 9, wherein said first interface call is parameterized by first data representing a first type of secure repository, and wherein said first and said second of said plurality of secure repositories are each of said first type.

16 (original). The method of claim 15, wherein said software process performs a second action if said one of said plurality of repositories is either said first or said second of said plurality of secure repositories, and wherein said software process does not perform said second action if said one of said plurality of secure repositories is a third of said plurality of secure repositories, said third of said plurality of secure repositories being of a second type different from said first type.

17 (original). The method of claim 9, further comprising the acts of:

dynamically linking to said software process a first set of computer-executable instructions, if said one of said plurality of repositories is said first of said plurality of secure repositories; and

dynamically linking to said software process a second set of computer-executable instructions different from said first set of computer-executable instructions, if said one of said plurality of secure repositories is said second of said plurality of secure repositories.

18 (original). The method of claim 9, further comprising the act of said software process receiving second data in response to said second interface call, said second data being generated by said one of said plurality of secure repositories, wherein said second data does not expose to said software process whether said data was generated by said first secure repository or said second secure repository.

19 (original). A computer-readable medium encoded with computer-executable instructions to perform the method of claim 9.

20 (currently amended). A secure repository comprising:

a first set of computer-executable instructions which converts encrypted data into decrypted data by applying a cryptographic key to said encrypted data without said cryptographic key being stored in any memory during the time that said first set of computer-executable instructions applies said cryptographic key; and

a second set of computer-executable instructions which provides said decrypted data to a software process if said secure repository trusts said software process;

wherein said secure repository establishes trust of said software process at least in part by establishing trust with an intermediate object, said intermediate object comprising a third set of computer-executable instructions dynamically linked to said software process.

21 (original). The secure repository of claim 20, wherein said software process renders said decrypted data.

22 (original). The secure repository of claim 20, further comprising a fourth set of computer-executable instructions which establishes trust with said intermediate object, said fourth set of computer-executable instructions including instructions to perform acts comprising:

receiving from said intermediate object first data comprising:

second data based at least in part on at least some code contained in said intermediate object; and

a signature of said second data; and

validating said signature.

23 (original). The secure repository of claim 22, wherein said second data comprises a hash of said at least some code.

24 (original). The secure repository of claim 22, wherein said fourth set of computer-executable instructions further performs acts comprising:

receiving from said intermediate object second data based at least in part on code contained in said software process.

25 (currently amended). A method of communicating with one of a plurality of secure repositories, said method comprising the acts of:

issuing a first interface call without regard to whether said one of said plurality of secure repositories is a first of said plurality of secure repositories or a second of said plurality of secure repositories;

if said one of said plurality of secure repositories is said first of said plurality of secure repositories, dynamically linking with a first set of computer-executable instructions invocable by said first interface call; and

if said one of said plurality of secure repositories is said second of said plurality of secure repositories, dynamically linking with a second set of computer-executable instructions invocable by said first interface call, said second said of computer-executable instructions being different from said first set of computer-executable instructions,

wherein said first of said plurality of secure repositories comprises a software module that uses a cryptographic algorithm to apply a cryptographic key to data without said cryptographic key being stored in a memory.

26 (currently amended). The method of claim 25, wherein each of said plurality of secure repositories converts encrypted data to decrypted data using [[a]] said cryptographic algorithm to apply [[a]] said cryptographic key to said encrypted data.

27 (original). The method of claim 25, wherein said first secure repository comprises a software-based secure repository, and wherein said second secure repository comprises at least some isolated hardware.

28 (original). The method of claim 25, wherein each of said first and second secure repositories are software-based repositories, said first secure repository having at least one feature not present in said second secure repository.

29 (cancelled).

30 (original). A computer-readable medium encoded with a second set of computer-executable instructions to perform the method of claim 25.

31 (currently amended). A method of authenticating a first software process to a second software process, said method comprising the acts of:

establishing to said second software process the authenticity of an intermediary object; and

using said intermediary object to establish to said second software process the authenticity of said first software process;

wherein said second software process converts encrypted data to decrypted data by using a cryptographic algorithm to apply a cryptographic key to said encrypted data without said cryptographic key being stored in a memory usable by said second software process during a time that said second software process is applying said cryptographic key, and wherein said first software process performs an operation on said decrypted data.

32 (cancelled).

33 (currently amended). The method of claim ~~32~~ 31, wherein said operation comprises rendering said decrypted data.

34 (original). The method of claim 33, wherein said first software process is a text-rendering application, and wherein said decrypted data comprises text.

35 (original). The method of claim 31, wherein said intermediary object comprises a set of computer-executable instructions having a first function callable from said first software process, and wherein the act of establishing to said second software process the authenticity of said intermediary object includes, or is actuated by, the act of said first software process calling said first function.

36 (original). The method of claim 35, wherein said act of establishing to said second software process the authenticity of said intermediary object includes the act of providing said second software process with a certificate based at least in part on said set of computer-executable instructions.

37 (original). The method of claim 36, wherein said certificate comprises a signed hash of at least some of said computer-executable instructions.

38 (original). The method of claim 35, wherein said intermediary object is in the address space of said first software process, and wherein said first function is referenceable by an address within said address space.

39 (original). The method of claim 35, wherein said set of computer-executable instructions is dynamically linkable with said first software process, and wherein said method further comprises the act of linking said set of computer-executable instructions with said first software process.

40 (original). The method of claim 31, wherein said intermediary object comprises a set of computer-executable instructions having a first function callable from said first software process, and wherein said act of using said intermediary object to establish to said second

software process the authenticity of said first software process includes, or is actuated by, the act of said first software process issuing a call to said first function.

a1

41 (original). A computer-readable medium encoded with a second set of computer-executable instructions to perform the method of claim 31.